



Implementation Specification for Registry Integration Module (RIM), version 1.0.0

October 2017

Federal Aviation Administration (FAA)
Single European Sky ATM Research (SESAR) Joint Undertaking (SJU)

Table of Contents

1	Scope.....	2
1.1	Scope	2
1.2	Introduction.....	2
1.3	Audience.....	3
1.4	Applicability	3
1.5	Notational Conventions.....	3
2	References.....	3
3	Architecture.....	4
3.1	Resources	5
3.2	Data Model	7
3.2.1	Logical Data Model.....	7
3.2.2	XML Schema.....	8
3.3	Binding.....	12
4	Interface.....	12
4.1	Request.....	13
4.2	Response	13
5	Operations.....	13
5.1	GetRegistry	13
5.2	GetIndex	14
5.3	GetServiceDescription	17
5.4	GetServiceProfile	20
6	Security	22
Appendixes	24
Appendix A	Web Application Description Language (WADL) Definition	24
Appendix B	Web Service Description Language (WSDL) Definition	26

1 Scope

1.1 Scope

This specification prescribes requirements for the development of the Registry Integration Module (RIM) for System Wide Information Management (SWIM) registries. This document specifies the interfaces, bindings, and a data model for defining application profiles required to publish and access services-related information in SWIM registries through machine-readable interfaces.

1.2 Introduction

Growing collaboration between international SWIM communities has presented new requirements for SWIM implementers, not just to exchange data among service components but also to exchange metadata including information about available services. To address these needs, the U.S. Federal Aviation Administration (FAA) and EUROCONTROL engaged in a joint effort to develop RIM, as a part of the Single European Sky ATM Research Programme (SESAR) Joint Undertaking (SJU) Coordination Plan 2.1.

RIM is a pluggable registry module whose goal is to enable dynamic exchange of service metadata among registries. RIM utilizes a well-known architectural paradigm, REST [[REST](#)], and a set of ubiquitous Web standards and protocols (e.g., URI, HTTP, XML).

Although RIM was designed to exchange service metadata between two SWIM registries, the SESAR European SWIM Registry and the FAA National Airspace System (NAS) Service Registry/Repository (NSRR), it also can be utilized by any other registry, or by any Web application supporting this specification.

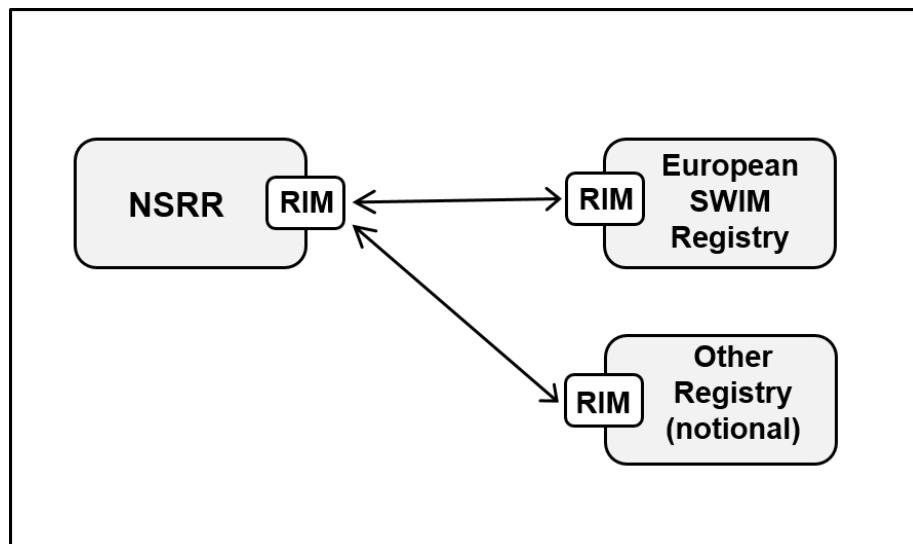


Figure 1 Integrating SWIM Registries by using RIM

1.3 Audience

The intended audience of this specification includes:

- Architects and developers designing, identifying, developing or maintaining service registries that will participate in national or international SWIM efforts.
- Standards architects and analysts developing specifications for finding or exchanging information about services (service metadata) maintained in SWIM service registries.
- Developers designing or creating Web applications for finding or exchanging service metadata maintained in SWIM service registries.
- Users who want a better understanding of the SWIM registry integration effort.

1.4 Applicability

This specification does not prescribe or limit the technological solutions for any registry. Rather, it seeks to provide a syntactic basis for building Web applications for finding or sharing services-related information in different SWIM registry implementations. This specification does not prescribe any configuration management (CM) or quality assurance (QA) policies, rules or assertions that a service registry may be subjected to. Finally, this specification does not prescribe any governance policies, rules or procedures that a service registry may be subjected to.

1.5 Notational Conventions

The keywords "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this document are to be interpreted as described in the IETF RFC 2119. When these words are not capitalized, they are meant in their natural-language sense.

2 References

[REST]	Architectural Styles and the Design of Network-based Software Architectures, Doctoral dissertation, Fielding, Roy Thomas, University of California, Irvine, 2000 http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
[UML]	Unified Modeling Language™ version 2.5, Object Management Group (OMG), 2015, http://www.omg.org/spec/UML/2.5/
[HTTP-RFC]	HTTP/1.1, RFC 2616, Fielding, Roy T., Tim Berners-Lee, et al., RFC Editor, 1999 http://www.rfc-editor.org/rfc/rfc2616.txt
[URI-RFC]	Uniform Resource Identifier (URI): Generic Syntax, RFC 3986, Berners-Lee, Tim, Roy T. Fielding, et al., RFC Editor, 2005

<http://www.rfc-editor.org/rfc/rfc3986.txt>

[SDCM]	Service Description Conceptual Model (SDCM) 2.0, June 2016 http://www.faa.gov/nextgen/programs/swim/governance/outreach/media/SDCM_v2.0/SDCM_v2.0.html
[XML-1.0]	Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26 November 2008 http://www.w3.org/TR/2008/REC-xml-20081126/
[WADL]	Web Application Description Language, W3C Member Submission 31 August 2009 http://www.w3.org/Submission/2009/SUBM-wadl-20090831/
[WSDL]	Web Service Description Language, W3C Recommendation 26 June 2007 https://www.w3.org/TR/wsdl20/
[HTTP-Auth]	HTTP Authentication: Basic and Digest Access Authentication, RFC 2617, J. Franks., et al, RFC Editor, 1999 https://tools.ietf.org/html/rfc2617
[OAuth]	The OAuth 2.0 Authorization Framework, RFC 6749, D. Hardt., RFC Editor, 2012 https://tools.ietf.org/html/rfc6749

3 Architecture

RIM deploys a number of architectural and technological approaches, including:

- Using HTTP as a transport protocol, the most interoperable protocol on the Web.
- Exposing information as resources that are uniquely identified using Web standards. Since service data are exchanged among multiple registries, such unique identification allows users to quickly identify the authoritative information source for each service.
- Supporting easy navigation. Links are provided between information resources so that a user can quickly discover information of interest, another key feature of the Web. For example, a registry may provide an index of all published services, and then allow the user to follow navigable links to individual service descriptions, even specific metadata elements within a service description.

RIM utilizes the Service Description Conceptual Model (SDCM) [[SDCM](#)] – another joint effort by EUROCONTROL and FAA – to describe relationships between elements of a service description and provide a common vocabulary for terms used within the model.

As depicted in the conceptual architecture in Figure 2, the RIM can be realized as a pluggable component that would provide the registry with additional functionalities without affecting current registry operations. By utilizing the existing registry's database as a data source, RIM can ensure the information's consistency across all interfaces provided by the registry.

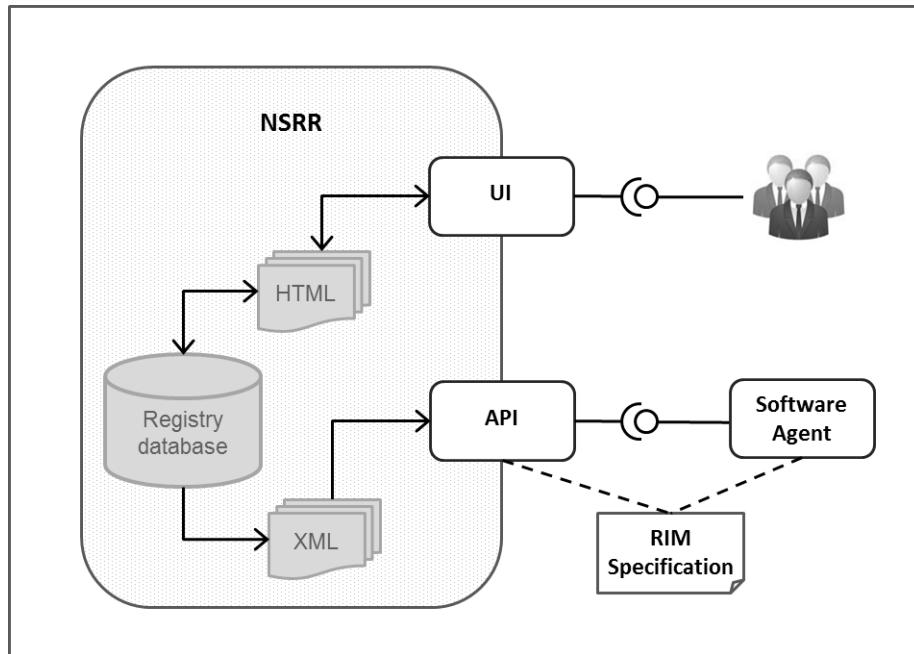


Figure 2 RIM Conceptual Architecture

3.1 Resources

The RIM interface is a collection of interlinked resources identified by URIs [[URI-RFC](#)]. Registries that implement the RIM specification SHALL assign a globally unique, HTTP-based URI called a *global registry identifier* (GRID) when a service is first published. Furthermore, the registry MUST ensure that the GRID is dereferenceable, i.e., a resource representation will be returned when an HTTP request is received with the GRID as the requested URI.

The RIM interface SHALL include the resources as listed in Table 1.

Table 1 RIM Resources

Resource URI template	Description

/registry	A resource representing the registry. It includes a basic description of the registry and contact information for the registry provider.
/index	A resource representing all the services published in the registry. It includes an index of all services registered.
/service-description?{GRID}	A resource representing the description of a service published in the registry. The registry must redirect an HTTP request addressed to the GRID to this resource.
/profile?{GRID}	A resource representing the profile of a service published in the registry. The registry must redirect an HTTP request addressed to the URI {GRID}/profile to this resource.

A registry MAY define additional resources. Furthermore, a registry MAY support additional query parameters for GET requests. For example, a registry may allow retrieval of registered entities modified after a certain date. Therefore, when a registry receives an HTTP request with query parameters it does not support, it MUST reject the request with HTTP response code 400. (See Section 4.2 below.)

All resources exposed by the RIM SHALL be interlinked to allow a user to traverse through the entire collection of resources.

The RIM resource model reflected in the UML class diagram [UML] in Figure 3 uses the following diagrammatic conventions:

- Each resource exposed through the RIM is modeled as a UML class with the stereotype “Resource”. The Resource stereotype has a property describing the resource’s URL pattern.
- Navigable links between resources are modeled as attributes or directed associations with the stereotype “locator”. The GRID is a locator.

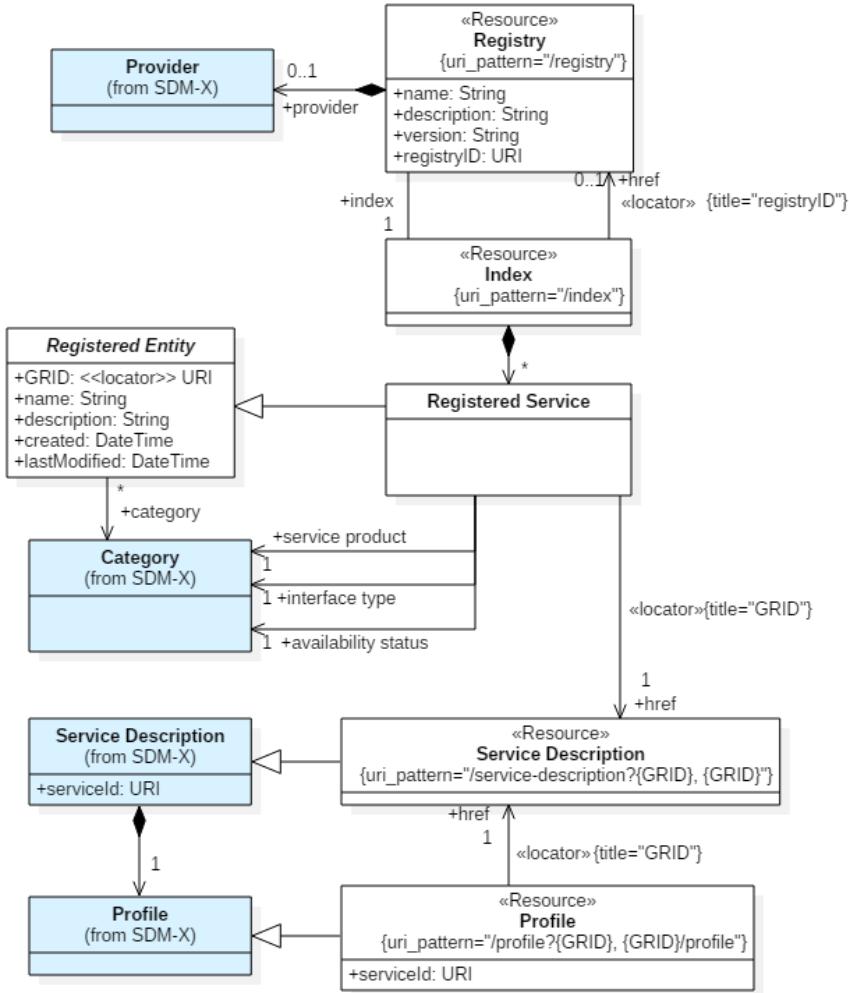


Figure 3 Resource Model

Web Application Description Language (WADL) [[WADL](#)] and Web Service Description Language (WSDL) [[WSDL](#)] definitions of these resources are included in Appendix A and Appendix B.

3.2 Data Model

3.2.1 Logical Data Model

The data model of the data accessible via the RIM has two components, as shown in Figure 4:

- A Service Description model that is based on the Service Description Conceptual Model (SDCM) [[SDCM](#)]. The SDCM does not suggest any technologies or implementation details but rather seeks to provide a common understanding that is expected to be used unambiguously between different implementations. This specification defines a data model that derives from the SDCM and supports the given architecture and resource model. Note: as a part of the initial implementation of the RIM, this data model only includes a subset of information defined in the SDCM.

- A Registry model that extends the Service Description model and defines the resources depicted in Figure 3.

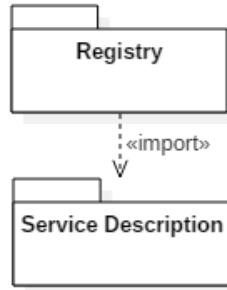


Figure 4 RIM Data Model

3.2.2 XML Schema

The XML Schema presented in this section serves as the physical data model for describing data elements transmitted via the RIM. Based on the logical model described in the previous section, the RIM defines two XML schemas:

- A registry schema that defines types and elements representing RIM resources, as shown below. The registry schema has the namespace “<http://swim.aero/rim/1.0.0>”.
- A service description schema (SDM-X) based on the SDCM. The service description schema has the namespace “<http://swim.aero/sdm-x/1.0.0>”.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:rim="http://swim.aero/rim/1.0.0"
  xmlns:sd="http://swim.aero/sdm-x/1.0.0"
  targetNamespace="http://swim.aero/rim/1.0.0" elementFormDefault="qualified"
  version="1.0.0">
  <xsd:annotation>
    <xsd:appinfo>
      <dc:identifier>http://swim.aero/rim/</dc:identifier>
      <dc:title> "Registry Interchange Model (RIM)"</dc:title>
      <dc:version>1.0.0</dc:version>
      <dc:description>This schema defines data elements to be used for
exchange of information among RIM-implementing registries.
      </dc:description>
      <dc:publisher>FAA/SESAR</dc:publisher>
      <dc:issued>2017-09-15</dc:issued>
      <dc:format>XML</dc:format>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:import namespace="http://swim.aero/sdm-x/1.0.0"
    schemaLocation=" http://swim.aero/sdm-x/1.0.0/service-description.xsd" />
  <xsd:import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="https://www.w3.org/1999/xlink.xsd" />

```

```

<xsd:element name="Registry" type="rim:RegistryType">
    <xsd:annotation>
        <xsd:documentation>
            This element represents an instance of a RIM-implementing
            registry.
            It is a root element for this schema.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:element name="Index" type="rim:IndexType">
    <xsd:annotation>
        <xsd:documentation>
            This element represents an indexed list of all entities registered
            -- defined by the RegisteredEntity element -- in the Registry.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:complexType name="RegistryType">
    <xsd:sequence>
        <xsd:element name="Name" type="xsd:string" />
        <xsd:element name="Description" type="xsd:string" />
        <xsd:element name="Version" type="xsd:string" />
        <xsd:element name="Provider" type="sd:ProviderType" />
        <xsd:element name="Index" type="rim:IndexType" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="registryId" type="xsd:anyURI" />
</xsd:complexType>

<xsd:complexType name="IndexType">
    <xsd:sequence>
        <xsd:element name="RegisteredService" type="rim:RegisteredServiceType"
                    minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute ref="xlink:type" fixed="locator" />
    <xsd:attribute ref="xlink:href" use="required" />
    <xsd:attribute ref="xlink:title" fixed="registryId" />
</xsd:complexType>

<xsd:complexType name="RegisteredEntityType" abstract="true">
    <xsd:annotation>
        <xsd:documentation>
            This abstract element represents the notion that different entities can
            be registered and subsequently presented by a RIM-implementing registry
            via the
            described interface. Although in this version
            only services are expected to be registered and presented, this element
            is established for future use.
            Name - the name of the entity.
            Description - a brief description of the entity.
            Category - one of the values from a predefined set of taxonomies.
            Created - the time and date when the entity was created or added to the
            registry.
            LastModified - the time and date of last update of the entity.
            GRID - the Global Registry Identifier that references the entity in the
            context of its hosting registry.
        </xsd:documentation>
    </xsd:annotation>

```

```

<xsd:sequence>
    <xsd:element name="Name" type="xsd:string" />
    <xsd:element name="Description" type="xsd:string" />
    <xsd:element name="Category" type="sd:CategoryType"
        minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="Created" type="xsd:dateTime" />
    <xsd:element name="LastModified" type="xsd:dateTime" />
</xsd:sequence>
<xsd:attribute ref="xlink:type" fixed="locator" />
<xsd:attribute ref="xlink:href" use="required" />
<xsd:attribute ref="xlink:title" fixed="GRID" />
</xsd:complexType>

<xsd:complexType name="RegisteredServiceType">
    <xsd:annotation>
        <xsd:documentation>
            This element represents a registered service in the context of a RIM
            registry. It extends RegistryEntityType by adding to the base class
            content a predefined set of values from three standard taxonomies.
            These taxonomies are designed to classify registered services and also
            are used to support a search via RIM.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="rim:RegisteredEntityType">
            <xsd:sequence>
                <xsd:element name="ServiceProductCategory"
                    type="rim:ServiceProductCategoryType" />
                <xsd:element name="AvailabilityStatusCategory"
                    type="rim:AvailabilityStatusCategoryType" />
                <xsd:element name="InterfaceTypeCategory"
                    type="rim:InterfaceTypeCategoryType" />
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ServiceProductCategoryType">
    <xsd:annotation>
        <xsd:documentation>
            A classification of services based on the type of SWIM data product that
            they deliver.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="sd:CategoryType">
            <xsd:sequence>
                <xsd:element name="Taxonomy" type="xsd:anyURI"
                    fixed="http://semantics.aero/service-product" />
                <xsd:element name="Code">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:anyURI">
                            <xsd:pattern
                                value="http://semantics.aero/service-product#[\S]+"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

</xsd:complexType>

<xsd:complexType name="InterfaceTypeCategoryType">
    <xsd:annotation>
        <xsd:documentation>
            A classification of services based on the type of technological solution
            that they deploy.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="sd:CategoryType">
            <xsd:sequence>
                <xsd:element name="Taxonomy" type="xsd:anyURI"
                    fixed="http://semantics.aero/interface-type" />
                <xsd:element name="Code">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:anyURI">
                            <xsd:pattern
                                value="http://semantics.aero/interface-type#[\S]+" />
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="AvailabilityStatusCategoryType">
    <xsd:annotation>
        <xsd:documentation>
            A classification of services based on their current, past, or future
            availability for provisioning.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="sd:CategoryType">
            <xsd:sequence>
                <xsd:element name="Taxonomy" type="xsd:anyURI"
                    fixed="http://semantics.aero/availability-status" />
                <xsd:element name="Code">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:anyURI">
                            <xsd:pattern
                                value="http://semantics.aero/availability-status#[\S]+" />
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ProfileType">
    <xsd:annotation>
        <xsd:documentation>
            This element represents a Profile resource by extending the
            ProfileType defined in the ServiceDescription schema,
            and adding a link to the Service Description resource.
        </xsd:documentation>
    </xsd:annotation>

```

```

<xsd:complexType>
    <xsd:extension base="sd:ProfileType">
        <xsd:attribute ref="xlink:type" fixed="locator" />
        <xsd:attribute ref="xlink:href" use="required" />
        <xsd:attribute ref="xlink:title" fixed="GRID" />
    </xsd:extension>
</xsd:complexType>
</xsd:complexType>

<xsd:element name="ServiceDescription" type="sd:ServiceDescriptionType">
    <xsd:annotation>
        <xsd:documentation>
            This element is fully defined in the ServiceDescription schema.
            It is re-declared in this schema only to make the described RIM API
            complete.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:element name="Profile" type="rim:ProfileType">
    <xsd:annotation>
        <xsd:documentation>
            Represents a Profile resource.
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:schema>

```

3.3 Binding

The RIM deploys a multilayer binding model typical for Web applications, with each layer binding to a specific protocol or a model.

Table 2 RIM Binding Model

<i>Layer</i>	<i>Protocol or Model</i>
Transport	Public Internet (TCP/IP)
Data	Data model as defined in section 3.2 of this specification.
Message	HTTP v.1.1 [HTTP-RFC]
Presentation	XML 1.0 [XML-1.0]/HTML 1.1 [HTTP-RFC]

4 Interface

The RIM inherently supports all aspects of HTTP including request methods, response codes and message headers. All messages consumed and produced by the RIM SHALL conform to the syntax defined in HTTP 1.1 [[HTTP-RFC](#)].

Sections 4.1 and 4.2 specify requirements for HTTP request and response messages respectively.

4.1 Request

The RIM MUST accept GET HTTP request messages as listed in Section 5.

The RIM MAY reject an HTTP request when the request header field ‘Accept’ does not contain the value “application/xml”.

4.2 Response

When the client request indicates an HTTP method not supported by the registry (for example “PUT” or “DELETE”), the RIM SHOULD respond with HTTP status code 405 “Method Not Allowed”. The response message SHALL include a header field ‘Allow’ with the value ‘GET’.

When a request message includes an “Accept” header field with values other than “application/xml”, the RIM SHALL return HTTP status code 406 “Not Acceptable”.

When a requested resource(s) is returned by the RIM, the response message SHALL include HTTP status code 200 “OK”. The response body SHALL include a representation of the requested resource as specified in Section 5.

When a resource name (resource URI) in an HTTP request line (e.g., GET /services/fps HTTP/1.1) is syntactically invalid, i.e., does not conform to one of the URI templates listed in Table 1, or the request includes an HTTP parameter that the registry does not support, the RIM SHALL respond with HTTP status code 400 “Bad Request”.

When a resource name supplied in the RIM HTTP request is syntactically valid, that is, compliant with the set of URI templates listed in Table 1, but no resource exists for a given URI (e.g., there is no service in the registry with a provided GRID), HTTP status code 404 “Not Found” SHALL be returned.

When a requested resource contains no data (e.g., Grounding may have no data entered in a registry in the early stages of a service’s development), HTTP status code 204 “No Content” SHALL be returned to signal that the response body is intentionally empty.

5 Operations

The RIM SHALL implement the operations described in this section.

5.1 GetRegistry

Description	Allows clients to retrieve basic information about the registry.
HTTP Method	GET

URI template	/registry
Obligation	Required
Input	None
Output	A representation of Registry object

Example of code:

```
# Request
GET /registry HTTP/1.1
Host: nsrr.faa.gov
Accept: application/xml

# Response
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<rim:Registry registryId="http://nsrr.faa.gov"
    xmlns:rim="http://swim.aero/rim/1.0.0"
    xmlns:sd="http://swim.aero//sdm-x/1.0.0"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <rim:Name>NSRR</rim:Name>
    <rim:Description>NAS Service Registry and Repository</rim:Description>
    <rim:Version>2.0.0</rim:Version>
    <rim:Provider>
        <sd:Name>Federal Aviation Administration (FAA) </sd:Name>
        <sd:Description>Maintained by the FAA SWIM program</sd:Description>
        <sd:WebPage>http://www.faa.gov</sd:WebPage>
        <sd:PointsOfContact>
            <sd:POC>
                <sd:Name>John Doe</sd:Name>
                <sd:Function>System Administrator</sd:Function>
                <sd:Phone>tel:+1-202-555-1212</sd:Phone>
                <sd:Email>admin@example.gov</sd:Email>
            </sd:POC>
        </sd:PointsOfContact>
    </rim:Provider>
</rim:Registry>
```

5.2 GetIndex

Description	Allows clients to retrieve an index of all registered services.
HTTP Method	GET
URI template	/index
Obligation	Required
Input	None

Output	A representation of Index object. The Index MUST include an XLink reference to the registry.
---------------	--

The GetIndex request has the following parameters:

Name	availability-status
Permissible Value	Values as defined in http://semantics.aero/availability-status Taxonomy
Optional	Yes

Name	service-product
Permissible Value	Values as defined in http://semantics.aero/service-product Taxonomy
Optional	Yes

Name	interface-type
Permissible Value	Values as defined in http://semantics.aero/interface-type Taxonomy
Optional	Yes

The taxonomy parameters in the GetIndex request can be specified by either using the full URI or just the last URI fragment. If the full URI is used for a taxonomy parameter, the registry MUST assume the URI has been encoded to escape special characters. In such cases, the registry MUST interpret the parameter by decoding it in accordance with [[URI-RFC](#)]. For example, `http%3A%2F%2Fswim.aero%2Fsemantics%2Favailability-status%23operational` specified in a GetIndex request will be decoded to `http://semantics.aero/availability-status#operational`.

The registry MUST return the same result in response to the following requests:

- GET /index?availability-status=http%3A%2F%2Fsemantics.aero%2Favailability-status%23operational
- GET /index?availability-status=operational

When a GetIndex request contains multiple parameters, it SHALL be interpreted as a logical AND. However, when multiple values are specified for the same parameter, they are interpreted as a logical

OR. Values for the same parameter are separated by ‘ , ’. For example, when a registry receives the following request:

- GET /index?availability-status=operational,retired&interface-type=resource-oriented

The response MUST include only services that have an availability status of “operational” OR “retired” AND an interface type of “resource-oriented”.

Example of code:

```
# Request
GET /index HTTP/1.1
Host: nsrr.faa.gov
Accept: application/xml
Authorization: Basic dXNlcjpwYXNzd29yZA==

# Response
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<rim:Index xmlns:rim="http://swim.aero/rim /1.0.0"
    xmlns:sd="http://swim.aero/sdm-x/1.0.0"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xlink:title="registryId "
    xlink:type="locator"
    xlink:href="https://nsrr.faa.gov">
    <rim:RegisteredService
        xlink:href="https://nsrr.faa.gov/services/fps"
        xlink:title="GRID"
        xlink:type="locator">
            <rim:Name>Flight Plan Service (FPS)</rim:Name>
            <rim:Description>A service for filing, updating, or canceling an IFR
                (Instrument Flight Rules) flight plan.</rim:Description>
            <rim:Created>2001-12-31T12:00:00</rim:Created>
            <rim:LastModified>2001-12-31T12:00:00</rim:LastModified>
            <sd:Taxonomy>
                <sd:Title>ATM Service Category</sd:Title>
                <sd:Value>Flight Planning</sd:Value>
            </sd:Taxonomy>
            <rim:ServiceProduct>
                <rim:Taxonomy>http://semantics.aero/service-product</rim:Taxonomy>
                <rim:Code>http://semantics.aero/service-product#flight</rim:Code>
            </rim:ServiceProduct>
            <rim:AvailabilityStatus>
                <rim:Taxonomy>http://semantics.aero/availability-status</rim:Taxonomy>
                <rim:Code>http://semantics.aero/availability-status#operational</rim:Code>
            </rim:AvailabilityStatus>
            <rim:InterfaceType>
                <rim:Taxonomy>http://semantics.aero/interface-type</rim:Taxonomy>
                <rim:Code>http://semantics.aero/interface-type#message-oriented</rim:Code>
            </rim:InterfaceType>
        </rim:RegisteredService>
    </rim:Index>
```

5.3 GetServiceDescription

Description	Allows clients to retrieve information about a registered service.
HTTP Method	GET
URI template	/service-description?{GRID}, where {GRID} is the global registry ID for the service being requested.
Obligation	Required
Input	None
Output	A representation of ServiceDescription object

This operation can also be invoked when the GRID is dereferenced. For example, the following URIs should cause the registry to return the same resource representation:

- <http://nsrr.faa.gov/service-description?http://nsrr.faa.gov/services/fps>
- <http://nsrr.faa.gov/services/fps>

Example of code:

```
# Request
GET /services/fps HTTP/1.1
Host: nsrr.faa.gov
Accept: application/xml
Authorization: Basic dXNlcjpwYXNzd29yZA==

# Response
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<rim:ServiceDescription
    xmlns:rim="http://swim.aero/rim/1.0.0"
    xmlns:sd="http://swim.aero/sdm-x/1.0.0"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <sd:Profile>
        <sd:ServiceName>Flight Plan Service (FPS)</sd:ServiceName>
        <sd:ServiceVersion>1.0.0</sd:ServiceVersion>
        <sd:ServiceDescription>(This fictitious service is for
            instructional use only and cannot be consumed) A service for filing,
            updating, or canceling an IFR (Instrument Flight Rules) flight plan.
        </sd:ServiceDescription>
        <sd:ServiceCategories>
            <sd:Category>
                <sd:CategoryName>ATM Service Category</sd:CategoryName>
                <sd:Value>Flight Planning</sd:Value>
            </sd:Category>
            <sd:Category>
                <sd:CategoryName>SWIM Service Product Category</sd:CategoryName>
                <sd:Value>Flight</sd:Value>
            </sd:Category>
            <sd:Category>
```

```

        <sd:CategoryName>Lifecycle Status</sd:CategoryName>
        <sd:Value>Definition</sd:Value>
    </sd:Category>
</sd:ServiceCategories>
<sd:Provider>
    <sd:Name>Federal Aviation Administration (FAA) </sd:Name>
    <sd>Description>Maintained by the FAA SWIM program</sd>Description>
    <sd:WebPage>http://www.faa.gov</sd:WebPage>
    <sd:PointsOfContact>
        <sd:POC>
            <sd:Name>John Doe</sd:Name>
            <sd:Function>System Administrator</sd:Function>
            <sd:Phone>tel:+1-202-555-1212</sd:Phone>
            <sd:Email>admin@example.gov</sd:Email>
        </sd:POC>
    </sd:PointsOfContact>
</sd:Provider>
<sd:Functions>
    <sd:Function>
        <sd>Description>File a flight plan.</sd>Description>
        <sd:RealWorldEffect>A flight plan has been filed and persists in
            the FAA Web server for distribution to the FAA flight data
            processing application within some parameter time of the
            estimated departure time.
        </sd:RealWorldEffect>
    </sd:Function>
    <sd:Function>
        <sd>Description>Cancel a flight plan.</sd>Description>
        <sd:RealWorldEffect>A previously filed flight plan has been
            retracted before being submitted to FAA Air Traffic Services,
            thereby reducing the flight plan processing load and systemic
            workload of the FAA air traffic planning system.
        </sd:RealWorldEffect>
    </sd:Function>
    <sd:Function>
        <sd>Description>Change destination aerodrome of a flight plan.
        </sd>Description>
        <sd:RealWorldEffect>The destination aerodrome of a filed flight
            plan has been changed.
        </sd:RealWorldEffect>
    </sd:Function>
</sd:Functions>
<sd:Security>
    <sd:SecurityMechanism>
        <sd:Name>Authorization</sd:Name>
        <sd>Description>The FPS deploys role-based access control (RBAC)
            for
            implementing authorization in accordance with ANSI/INCITS
            359-2004.
            Two roles are defined, “Reader” (a user who only has
            permission to
            view a filed flight plan) and “Originator” (a user,
            generally a
            pilot or operator, who submits a flight plan and has
            permission to
            file and subsequently modify or cancel the filed
            flight plan).
        </sd>Description>
        <sd:RegulatingProtocol>
            <sd>Title>Role Based Access Control</sd>Title>
        </sd:RegulatingProtocol>
    </sd:SecurityMechanism>
</sd:Security>
<sd:Policies>
    <sd:Policy>

```

```

        <sd:Title>Flight Plan Service (FPS) Policy Document Version 1
        </sd:Title>
        <sd:Location>https://www.faa.gov/atm/policies/fps-
policy.xml</sd:Location>
            </sd:Policy>
            <sd:Policy>
                <sd:Title>Runtime Policy</sd:Title>
            <sd:Location>http://www.faa.gov/air_traffic/flight_info/operation_policy.pdf</sd:Location>
                </sd:Policy>
                <sd:Policy>
                    <sd:Title>FAA Order 1370.92A, Password and PIN Management Policy
                    </sd:Title>

                <sd:Location>http://www.faa.gov/documentLibrary/media/Order/1370.92A.pdf</sd:Location>
                    </sd:Policy>
                    <sd:Policy>
                        <sd:Title>NIST FIPS Publication 200, Minimum Security
                            Requirements for
                            Federal Information and Information Systems
                        </sd:Title>
                        <sd:Location>http://csrc.nist.gov/publications/fips/fips200/FIPS-200-
final-march.pdf</sd:Location>
                            </sd:Policy>
                    </sd:Policies>
                    <sd:QualitiesOfService>
                        <sd:QualityOfServiceParameter>
                            <sd:Name>Capacity</sd:Name>
                            <sd:Value>20 per minute</sd:Value>
                            <sd:Definition>Number of service requests that the service can
                                accommodate within a given time period.
                            </sd:Definition>
                            <sd:CalculationMethod>Simple count.
                            </sd:CalculationMethod>
                            <sd:UnitOfMeasure>Whole positive number, per period of time
                            </sd:UnitOfMeasure>
                        </sd:QualityOfServiceParameter>
                        <sd:QualityOfServiceParameter>
                            <sd:Name>Availability</sd:Name>
                            <sd:Value>≥ 99.900</sd:Value>
                            <sd:Definition>Probability that the service is present or ready
                                for immediate use.
                            </sd:Definition>
                            <sd:CalculationMethod>100 * ((24 - Total Outage Time) / 24).
                                Measurements are taken daily and apply to the preceding 24-hour
                                period.
                            </sd:CalculationMethod>
                            <sd:UnitOfMeasure>Percentage, accurate to 3 decimal places
                            </sd:UnitOfMeasure>
                        </sd:QualityOfServiceParameter>
                    </sd:QualitiesOfService>
                    <sd:EnvironmentalConstraints>
                        <sd:Constraint>
                            <sd:Description>The FPS operates within the FAA
                                Telecommunications Infrastructure (FTI) and is subject to its
                                performance constraints. All FPS requests and responses are
                                brokered via the NAS Enterprise Messaging Service (NEMS).
                            </sd:Description>
                        </sd:Constraint>
                    </sd:EnvironmentalConstraints>
                </sd:Profile>
                <sd:Model/>
                <sd:Grounding/>
            </rim:ServiceDescription>

```

5.4 GetServiceProfile

Description	Allows clients to retrieve profile information about a registered service.
HTTP Method	GET
URI template	/profile?{GRID}, where {GRID} is the global registry ID for the service being requested.
Obligation	Required
Input	None
Output	A representation of Profile object.

This operation can also be invoked when the {GRID}/profile is dereferenced. For example, the following URLs should cause the registry to return the same resource representation:

- <http://nsrr.faa.gov/profile?http://nsrr.faa.gov/services/fps/profile>
- <http://nsrr.faa.gov/services/fps/profile>

Example of code:

```
# Request
GET /services/fps/profile HTTP/1.1
Host: nsrr.faa.gov
Accept: application/xml
Authorization: Basic dXNlcjpwYXNzd29yZA==

# Response
HTTP/1.1 200 OK
Content-Type: application/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<rim:Profile
    xlink:href="http://nsrr.faa.gov/services/fps"
    xlink:title="GRID"
    xlink:type="locator"
    xmlns:rim="http://swim.aero/rim/1.0.0"
    xmlns:sd="http://swim.aero/sdm-x/1.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <sd:ServiceName>Flight Plan Service (FPS)</sd:ServiceName>
    <sd:ServiceVersion>1.0.0</sd:ServiceVersion>
    <sd:ServiceDescription>(This fictitious service is for
        instructional use only and cannot be consumed) A service for filing,
        updating, or canceling an IFR (Instrument Flight Rules) flight plan.
    </sd:ServiceDescription>
    <sd:ServiceCategories>
        <sd:Category>
            <sd:CategoryName>ATM Service Category</sd:CategoryName>
            <sd:Value>Flight Planning</sd:Value>
        </sd:Category>
        <sd:Category>
            <sd:CategoryName>SWIM Service Product Category</sd:CategoryName>
            <sd:Value>Flight</sd:Value>
        </sd:Category>
    </sd:ServiceCategories>
</rim:Profile>
```

```

        <sd:Category>
            <sd:CategoryName>Lifecycle Status</sd:CategoryName>
            <sd:Value>Definition</sd:Value>
        </sd:Category>
    </sd:ServiceCategories>
    <sd:Provider>
        <sd:Name>Federal Aviation Administration (FAA) </sd:Name>
        <sd>Description>Maintained by the FAA SWIM program</sd>Description>
        <sd:WebPage>http://www.faa.gov</sd:WebPage>
        <sd:PointsOfContact>
            <sd:POC>
                <sd:Name>John Doe</sd:Name>
                <sd:Function>System Administrator</sd:Function>
                <sd:Phone>tel:+1-202-555-1212</sd:Phone>
                <sd>Email>admin@example.gov</sd>Email>
            </sd:POC>
        </sd:PointsOfContact>
    </sd:Provider>
    <sd:Functions>
        <sd:Function>
            <sd>Description>File a flight plan.</sd>Description>
            <sd:RealWorldEffect>A flight plan has been filed and persists in
                the FAA Web server for distribution to the FAA flight data
                processing application within some parameter time of the
                estimated departure time.
            </sd:RealWorldEffect>
        </sd:Function>
        <sd:Function>
            <sd>Description>Cancel a flight plan.</sd>Description>
            <sd:RealWorldEffect>A previously filed flight plan has been
                retracted before being submitted to FAA Air Traffic Services,
                thereby reducing the flight plan processing load and systemic
                workload of the FAA air traffic planning system.
            </sd:RealWorldEffect>
        </sd:Function>
        <sd:Function>
            <sd>Description>Change destination aerodrome of a flight plan.
            </sd>Description>
            <sd:RealWorldEffect>The destination aerodrome of a filed flight
                plan has been changed.
            </sd:RealWorldEffect>
        </sd:Function>
    </sd:Functions>
    <sd:Security>
        <sd:SecurityMechanism>
            <sd:Name>Authorization</sd:Name>
            <sd>Description>The FPS deploys role-based access control (RBAC)
                For implementing authorization in accordance with ANSI/INCITS
                359-2004. Two roles are defined, “Reader” (a user who only has
                permission to view a filed flight plan) and “Originator” (a user,
                generally a pilot or operator, who submits a flight plan and has
                permission to file and subsequently modify or cancel the filed
                flight plan).
            </sd>Description>
            <sd:RegulatingProtocol>
                <sd>Title>Role Based Access Control</sd>Title>
            <sd:Location>http://www.cs.purdue.edu/homes/ninghui/readings/AccessControl/ANSI+INCITS+359-2004.pdf</sd:Location>
                </sd:RegulatingProtocol>
            </sd:SecurityMechanism>
        </sd:Security>
        <sd:Policies>
            <sd:Policy>
                <sd>Title>Flight Plan Service (FPS) Policy Document Version 1
                </sd>Title>
                <sd:Location>https://www.faa.gov/atm/policies/fps-policy.xml</sd:Location>
            </sd:Policy>
        </sd:Policies>
    </sd:Service>

```

```

        <sd:Policy>
            <sd:Title>Runtime Policy</sd:Title>
            <sd:Location>http://www.faa.gov/air\_traffic/flight\_info/operation\_policy.pdf</sd:Location>
        </sd:Policy>
        <sd:Policy>
            <sd:Title>FAA Order 1370.92A, Password and PIN Management Policy</sd:Title>
            <sd:Location>http://www.faa.gov/documentLibrary/media/Order/1370.92A.pdf</sd:Location>
        </sd:Policy>
        <sd:Policy>
            <sd:Title>NIST FIPS Publication 200, Minimum Security Requirements for
                    Federal Information and Information Systems</sd:Title>
            <sd:Location>http://csrc.nist.gov/publications/fips/fips200/FIPS-200-final-march.pdf</sd:Location>
        </sd:Policy>
    </sd:Policies>
    <sd:QualitiesOfService>
        <sd:QualityOfServiceParameter>
            <sd:Name>Capacity</sd:Name>
            <sd:Value>20 per minute</sd:Value>
            <sd:Definition>Number of service requests that the service can
                accommodate within a given time period.
            </sd:Definition>
            <sd:CalculationMethod>Simple count.
            </sd:CalculationMethod>
            <sd:UnitOfMeasure>Whole positive number, per period of time
            </sd:UnitOfMeasure>
        </sd:QualityOfServiceParameter>
        <sd:QualityOfServiceParameter>
            <sd:Name>Availability</sd:Name>
            <sd:Value>≥ 99.900</sd:Value>
            <sd:Definition>Probability that the service is present or ready
                for immediate use.
            </sd:Definition>
            <sd:CalculationMethod>100 * ((24 - Total Outage Time) / 24).
                Measurements are taken daily and apply to the preceding 24-hour
                period.
            </sd:CalculationMethod>
            <sd:UnitOfMeasure>Percentage, accurate to 3 decimal places
            </sd:UnitOfMeasure>
        </sd:QualityOfServiceParameter>
    </sd:QualitiesOfService>
    <sd:EnvironmentalConstraints>
        <sd:Constraint>
            <sd:Description>The FPS operates within the FAA
                Telecommunications Infrastructure (FTI) and is subject to its
                performance constraints. All FPS requests and responses are
                brokered via the NAS Enterprise Messaging Service (NEMS).
            </sd:Description>
            </sd:Constraint>
    </sd:EnvironmentalConstraints>
</sd:Profile>
```

6 Security

A registry should establish a single set of user identities across the RIM and its Web-based interface, and enforce the same authentication and access control policies across all interfaces. It is strongly recommended that the RIM require authentication and employ access control mechanisms when a client invokes an operation that will modify the content of the registry.

When security is required, the registry will be expected to use an authentication method that is compatible with the HTTP protocol, such as HTTP basic or digest authentication [[HTTP-Auth](#)], or a token-based authentication mechanism such as the OAuth 2.0 Authorization Framework [[OAuth](#)].

An authentication token MUST be included in the HTTP Authorization header. If no HTTP Authorization header is included in the request, the RIM SHALL respond with HTTP response code 401 “Unauthorized”. If the client is authenticated but not allowed to access the resource requested, the RIM MAY respond with the HTTP response code 403 “Forbidden”.

Appendices

Appendix A Web Application Description Language (WADL) Definition

```
<application xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://wadl.dev.java.net/2009/02 https://www.w3.org/Submission/wadl/wadl.xsd"
    xmlns:tns="http://swim.aero/rim/1.0.0" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:rim="http://swim.aero/rim/1.0.0" xmlns="http://wadl.dev.java.net/2009/02">
    <doc xml:lang="en">
        Registry Integration Module (RIM) 1.0.0
    </doc>
    <grammars>
        <!-- RIM Schema -->
        <include href="http://swim.aero/rim/1.0.0/rim-1.0.0.xsd" />
    </grammars>

    <resource_type id="Registry">
        <method name="GET" id="GetRegistry">
            <doc>Returns basic information about the Registry</doc>
            <request />
            <response status="200">
                <representation mediaType="application/xml" element="rim:Registry" />
            </response>
            <response status="406">
                <doc>Not acceptable: if the client has requested a media
                    type that is not supported.
                </doc>
            </response>
        </method>
    </resource_type>
    <resource_type id="Index">
        <method name="GET" id="GetIndex">
            <doc>Returns an index of registered entities within the Registry
            </doc>
            <request>
                <doc>The request may include optional parameters to filter the
                    result based on common taxonomies
                </doc>
                <param name="interface_type" style="query" type="xsd:NCName"
                    repeating="true" required="false">
                    <doc>Example: request only resource oriented OR message oriented
                        services
                        /index?interface_type=resource_oriented,message_oriented
                    </doc>
                </param>
                <param name="availability_status" style="query" type="xsd:NCName"
                    repeating="true" required="false" />
                <param name="service_product" style="query" type="xsd:NCName"
                    repeating="true" required="false" />
            </request>
            <response status="200">
                <representation mediaType="application/xml" element="rim:Registry" />
            </response>
            <response status="406">
                <doc>Not acceptable: if the client has requested a media
                    type that is not supported.
                </doc>
            </response>
        </method>
    </resource_type>
    <resource_type id="ServiceDescription">
        <method name="GET" id="GetServiceDescription">
```

```

<doc>Returns the representation of a Service Description</doc>
<request />
<response status="200">
    <representation mediaType="application/xml"
element="rim:ServiceDescription" />
    </response>
    <response status="404">
        <doc>Not found: if the client has requested a service that does not
            exist.
        </doc>
    </response>
    <response status="406">
        <doc>Not acceptable: if the client has requested a media
            type that is not supported.
        </doc>
    </response>
</method>
</resource_type>
<resource_type id="ServiceProfile">
    <method name="GET" id="GetServiceProfile">
        <doc>Returns the representation of a Service Profile</doc>
        <request />
        <response status="200">
            <representation mediaType="application/xml" element="rim:Profile" />
        </response>
        <response status="404">
            <doc>Not found: if the client has requested a service that does not
                exist.
            </doc>
        </response>
        <response status="406">
            <doc>Not acceptable: if the client has requested a media
                type that is
                not supported.
            </doc>
        </response>
    </method>
</resource_type>
</application>

```

Appendix B Web Service Description Language (WSDL) Definition

```

<wsdl:description xmlns:wsdl="http://www.w3.org/ns/wsdl"
    targetNamespace="http://swim.aero/rim/1.0.0"
    xmlns:whttp="http://www.w3.org/ns/wsdl/http"
    xmlns:wsdlx="http://www.w3.org/ns/wsdl-extensions"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rim="http://swim.aero/rim/1.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <wsdl:documentation>Registry Integration Module (RIM) 1.0.0</wsdl:documentation>
    <wsdl:types>
        <xs:import namespace="http://swim.aero/rim/1.0.0"
            schemaLocation="http://swim.aero/rim/1.0.0/rim-1.0.0.xsd" />
        <xs:schema targetNamespace="http://swim.aero/rim/1.0.0"
            xmlns:rim="http://swim.aero/rim/1.0.0"
            xmlns:xs="http://www.w3.org/2001/XMLSchema">
            <xs:complexType name="QueryParameters">
                <!-- optional query parameters -->
                <xs:sequence>
                    <xs:element name="interface_type" type="xs:NCName"
                        minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="availability_status" type="xs:NCName"
                        minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="service_product" type="xs:NCName"
                        minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
            <xs:element name="queryParameters" type="rim:QueryParameters"/>
            <xs:element name="GRID" type="xs:anyURI">
                <!-- An HTTP URL assigned by the registry to the Service Description -->
                <xs:element name="lid" type="xs:NCName">
                    <!-- the local part of GRID -->
                </xs:element>
            </xs:schema>
        </wsdl:types>

        <wsdl:interface name="RIM">
            <wsdl:operation name="GetRegistry" pattern="http://www.w3.org/ns/wsdl/in-out"
                style="http://www.w3.org/ns/wsdl/style/iri" wsdlx:safe="true">
                <wsdl:documentation>
                    This operation returns a description of the registry.
                </wsdl:documentation>
                <wsdl:input/>
                <wsdl:output element="rim:Registry" />
            </wsdl:operation>
            <wsdl:operation name="GetIndex" pattern="http://www.w3.org/ns/wsdl/in-out"
                style="http://www.w3.org/ns/wsdl/style/iri" wsdlx:safe="true">
                <wsdl:documentation>
                    This operation returns a list of registered entities.
                </wsdl:documentation>
                <wsdl:input element="rim:queryParameters"/>
                <wsdl:output element="rim:Index" />
            </wsdl:operation>
            <wsdl:operation name="GetServiceDescription" pattern="http://www.w3.org/ns/wsdl/in-out"
                style="http://www.w3.org/ns/wsdl/style/iri" wsdlx:safe="true">
                <wsdl:documentation>
                    This operation returns a representation of service description
                </wsdl:documentation>
                <wsdl:input element="rim:GRID"/>
                <wsdl:output element="rim:ServiceDescription" />
            </wsdl:operation>
            <wsdl:operation name="GetServiceProfile"
                pattern="http://www.w3.org/ns/wsdl/in-out"
                style="http://www.w3.org/ns/wsdl/style/iri" wsdlx:safe="true">
                <wsdl:documentation>
                    This operation returns a representation of service profile
                </wsdl:documentation>
            </wsdl:operation>
        </wsdl:interface>
    </wsdl:description>

```

```
</wsdl:documentation>
    <wsdl:input element="rim:GRID"/>
    <wsdl:output element="rim:ServiceDescription" />
</wsdl:operation>
</wsdl:interface>

<wsdl:binding name="RIM_REST"
    type="http://www.w3.org/ns/wsdl/http"
    interface="rim:RIM">
    <wsdl:documentation>
        The RESTful HTTP binding for RIM.
    </wsdl:documentation>
    <wsdl:operation ref="GetIndex"
        whttp:location="/index/"
        whttp:method="GET"
        whttp:outputSerialization="text/xml" />
    <wsdl:operation ref="GetRegistry"
        whttp:location="/registry/"
        whttp:method="GET"
        whttp:outputSerialization="text/xml" />
    <wsdl:operation ref="GetServiceDescription"
        whttp:location="/service-description?{GRID}"
        whttp:method="GET"
        whttp:outputSerialization="text/xml" />
    <wsdl:operation ref="GetServiceProfile"
        whttp:location="/profile?{GRID}"
        whttp:method="GET"
        whttp:outputSerialization="text/xml" />
</wsdl:binding>
</wsdl:description>
```